Manuscript for ArXiv.org

The IBMAP approach for Markov networks structure learning

Federico Schlüter · Facundo Bromberg · Alejandro Edera

January 17, 2013

Abstract In this work we consider the problem of learning the structure of Markov networks from data. We present an approach for tackling this problem called IBMAP, together with an efficient instantiation of the approach: the IBMAP-HC algorithm, designed for avoiding important limitations of existing independence-based algorithms. These algorithms proceed by performing statistical tests of independence on data, trusting completely the outcome of each test. In practice tests may be incorrect, resulting in potential cascading errors and the consequent reduction in the quality of the structures learned. IBMAP contemplates this uncertainty in the outcome of the tests through a probabilistic maximum-a-posteriori approach. The approach is instantiated in the IBMAP-HC algorithm, a structure selection strategy that performs a polynomial heuristic local search in the space of possible structures. We present an extensive empirical evaluation on synthetic and real data, showing that our algorithm outperforms significantly the existent independence-based algorithms, in terms of data efficiency and quality of learned structures, with equivalent computational complexities. We also show the performance of IBMAP-HC in a real-world application of knowledge discovery: EDAs, which are evolutive algorithms that use structure learning on each generation for modeling the distribution of populations. The experiments show that when IBMAP-HC is used to learn the structure, EDAs improve the convergence to the optimum.

Keywords Markov networks · Structure learning · independence tests · knowledge discovery · EDAs

F. Schlüter, F. Bromberg, A. Edera
Lab. DHARMa of Artificial Intelligence,
Departamento de Sistemas de información,
Facultad Regional Mendoza, Universidad Tecnológica Nacional, Argentina.
Tel.: +54-261-5244566
E-mail: {federico.schluter,fbromberg,aedera}@frm.utn.edu.ar

1 Introduction

We present in this work the IBMAP (Independence-Based Maximum a Posteriori) approach for robust learning of Markov network structures from data, together with IBMAP-HC, an efficient hill-climbing instantiation of the approach. Markov networks, together with Bayesian networks, belong to the family of *probabilistic graphical models* [16], a computational framework for compact representation of joint probability distributions. They consist of an undirected (Markov networks) or directed (Bayesian networks) graph G and a set of numerical parameters θ. Each node in G represents a random variable of the domain, and the edges encode conditional independences between them. Therefore, the graph G is called the *independence structure* of the distribution. The importance of these independences is that they factorize the joint distribution over the domain variables into factors over subsets of variables, resulting in important reductions in the space complexity required to store a distribution [32].

An interesting problem is to represent the distribution of historical information in a Markov network. This task can be done either by a human expert, or by learning it automatically from data. Naturally, the knowledge of experts is hard to obtain, and not always enough to design a proper Markov network. For this reason, algorithms for learning automatically such models are considered an increasingly important tool for knowledge discovery. The problem consists in two tasks, learning G and learning θ. In this work we focus in the problem of learning G, that has shown to be an NP-hard problem [10] since the number of structures grows super-exponentially with the number of variables of the domain.

The literature considers two broad approaches for learning G: *score-based* [13, 22, 19, 14], and *independence-based* (also known as constraint-based) algorithms [32]. The score-based approach is intractable in practice for large domains for two reasons: (i) it approaches the problem as an optimization on the space of G, and (ii) for each G during the search it requires a learning step of the model parameters θ, which involves an expensive inference step [34]. The independence-based approach proceeds by performing statistical tests of independence on the data, and based on the outcome of the tests discards all structures inconsistent with the test. This approach is efficient, and correct under assumptions, but in practice presents quality problems: one of the assumptions is the correctness of independence assertions, which may not be true in practice for statistical tests when data is insufficient. These problems are described in detail in a recently published survey [29]. It is important to mention that both score-based and independence-based approaches have been motivated by distinct learning goals. Generally, score-based approaches are better suited for the density estimation goal, that is, tasks where inferences or predictions are required [23]. In contrast, independence-based methods are better suited for other learning goals, such as feature selection for classification, or knowledge discovery [32, 4, 5].

IBMAP follows the independence-based approach but relaxes the assumption of correctness of independence assertions. Instead of trusting the outcome of a statistical test on data, it considers explicitly the posterior probability of independences given the data. As explained in detail later on, these posteriors are combined into the posterior of the whole structure (given the data), deciding on the output structure following the well-known maximum-a-posteriori approach. This clearly circumvents the cascading error, as the true structure is no longer dis-

carded on an incorrect test, it only results in a lower posterior probability. With further tests, the posterior probability of the true structure may increase again.

In order to evaluate the improvements in the quality of the structures produced by our approach, we performed detailed and systematic experiments over synthetic datasets, real-world datasets. In all those cases we compared the structural errors of the structures learned by IBMAP-HC and against those learned by representative, state-of-the-art competitors: GSMN [8,9], and HHC-MN, a simple adaptation for Markov networks of an independence-based structure learning algorithm for Bayesian networks, called HHC [5]. We note that structural errors as quality measure is the most appropriate for knowledge discovery algorithms such as those using the independence-based approach.

Additionally we tested the performance of IBMAP-HC in a real world application: *Estimation of Distribution algorithms* (EDAs) [25,17]. These algorithms are variations of the well-known evolutionary algorithms, that replace the crossover and mutation stages for generating a new population, by learning a probability distribution of the current population. This application is relevant because EDAs solve problems that are known to be hard for traditional genetic algorithms. We tested IBMAP-HC in a state-of-the-art EDA algorithm based on Markov networks structure learning, called the MOA algorithm [31]. In our experiments, MOA improves its convergence to the optimum when IBMAP-HC is used to learn the structure.

The rest of this work is organized as follows. Section 2 presents an overview of the independence-based learning approach and motivates our contribution. Section 3 presents IBMAP and the IBMAP-HC algorithm. Section 4 shows our experiments on synthetic and real datasets, and Section 5 shows our experiments on EDAs. Finally, Section 6 summarizes this work, and poses several possible directions of future work.

2 Background

This section provides some background on Markov networks, defines the problem of structure learning, and motives our independence-based approach.

A Markov network representing an underlying distribution $P(\mathbf{V})$ over the set of $n = |\mathbf{V}|$ random variables $\mathbf{V}$ consists in an undirected graph G and a set of potential functions defined by a set of numerical parameters Θ. The graph G is a map of the independences in $P(\mathbf{V})$, and such independences can be read from the graph through *vertex separation*, considering that each variable is conditionally independent of all its non-neighbor variables in the graph, given the set of its neighbor variables [26].

The structure G of $P(\mathbf{V})$ can be factorized into a product of *potential* functions $\phi_c(V_c)$ over the completely connected sub-graphs (a.k.a., *cliques*) V_c of G, that is,

$$P(\mathbf{V}) = \frac{1}{Z} \prod_{c \in cliques(G)} \phi_c(V_c),$$

where Z is the *partition function*, a constant that normalizes the product of potentials. Such potential functions are parameterized by the set of numerical parameters Θ.

The problem of structure learning takes as input a dataset D, which is assumed to be a representative sample of the underlying distribution $P(\mathbf{V})$. Commonly, D is structured in a tabular format, with one column per random variable in the domain $\mathbf{V}$, and one row per data point. The optimal solution of the problem is a perfect-map of $P(\mathbf{V})$ [26], that is, a structure that encodes all the dependences and the independences present in $P(\mathbf{V})$. The closer to a perfect-map, the better is the structure learned, and the better is the resulting Markov network for representing $P(\mathbf{V})$.

Independence-based algorithms learn a perfect-map by performing a succession of statistical independence tests, discarding at each iteration all structures inconsistent with the outcome of the test, and deciding on the tests to perform next based on the outcomes learned so far.

A statistical independence test is a statistics computed from D which tests if two random variables X and Y are conditionally independent, given some conditioning set of variables $\mathbf{Z}$. This *independence assertion* is denoted $\langle X \perp\!\!\!\perp Y | \mathbf{Z} \rangle$ (or $\langle X \not\perp\!\!\!\perp Y | \mathbf{Z} \rangle$ for *dependence assertions*). The computational cost of a test is proportional to the number of rows in D, and the number of variables involved in the test. Examples of independence tests used in practice are Mutual Information [11], Pearson's χ^2 and G^2 [2], the Bayesian test [20], and for continuous Gaussian data the *partial correlation* test [32].

There are several advantages of independence-based algorithms. First, they can learn the structure without interleaving the expensive task of parameters estimation (contrary to score-based algorithms), reaching sometimes polynomial complexities in the number of statistical tests performed. If the complete model is required, the parameters can be estimated only once for the learned structure. Another important advantage of such algorithms is that they are guaranteed to learn the structure of the underlying distribution, as long as the following assumptions hold: a) *graph-isomorphism* (i.e., the independences in the distribution can be encoded in an undirected graph), b) the underlying distribution is *strictly positive* (i.e., $P(\mathbf{V}) > 0$, for every assignment of $\mathbf{V}$), and c) the outcomes of tests are correct (i.e., the independencies learned are true in $P(\mathbf{V})$).

Unfortunately, the third assumption is rarely true in practice, as the number of contingency tables for which a statistics has to be computed grows exponentially with the number of variables in the conditioning set of the test. Therefore, the effective dataset from which the statistic is computed decreases exponentially in size, thus degrading exponentially the quality of the statistics. When tests outcome incorrect independences, independence-based algorithms produce what is commonly called *cascade errors* [32], that not only discard the true underlying structure, but further confuse the algorithm in the test to perform next. Our approach tackles this main issue of independence-based algorithms by contemplating the uncertainty in the outcome of the tests through a probabilistic maximum-a-posteriori approach.

3 The independence-based MAP approach

We describe now the main contribution of this work: the IBMAP approach for Markov networks structure learning. The central idea is to aggregate the result of many statistical tests of conditional independence into the posterior probability $P(G \mid D)$ of the independence structure G, and, by taking the maximum-a-

posteriori (MAP) model selection approach, selecting as output the structure that maximizes this posterior. Formally

$$G^\star = \arg\max_G \Pr(G \mid D). \tag{1}$$

We proceed now to discuss how the individual posteriors $\Pr(G \mid D)$ are computed, and later in Section 3.1 how to perform the MAP optimization efficiently. We start by replacing G in $\Pr(G \mid D)$ by the *closure* $\mathcal{C}(G)$, an equivalent representation of G consisting of a set of independence assertions that determines it. Formally

Definition 1 (Closure) The *closure* of an undirected independence structure G is a set of conditional independence assertions, $\mathcal{C}(G) = \{c_i\}$, that are sufficient for completely determining the structure G of a positive distribution.

We thus have that

$$\Pr(G \mid D) = \Pr(\mathcal{C}(G) \mid D).$$

Applying the chain rule over the independence assertions in $\mathcal{C}(G)$ we obtain:

$$\Pr(\mathcal{C}(G) \mid D) = \prod_{i=1}^{|\mathcal{C}(G)|} \Pr(c_i | c_1, \ldots, c_{i-1}, D). \tag{2}$$

To the best of the author's knowledge, there is no existing method for computing exactly the probabilities $\Pr(c_i|c_1, \ldots, c_{i-1}, D)$ of independence assertions conditioned on other independence assertions and data. A common approximation is to assume that all the independence assertions in the closure *are mutually independent*. This assumption is made implicitly by all the Markov networks independence-based algorithms [29], because the statistical tests are used as a black box, only using data for deciding independence for each assertion c_i. Applying the approximation to Eq. (2) we obtain

$$\Pr(\mathcal{C}(G) \mid D) \approx \prod_i \Pr(c_i \mid D)$$

that expressed in terms of logarithms to avoid underflow results in a computable expression that we call the *IB-score*:

$$\sigma(G) = \sum_i \log \Pr(c_i \mid D), \tag{3}$$

where each term $\log \Pr(c_i \mid D)$ can be computed using the Bayesian test of conditional independence [20].

In summary, the IBMAP approach consists in the following maximization:

$$G^\star \approx \arg\max_G \ \sigma(G). \tag{4}$$

3.1 The IBMAP-HC algorithm

This section present our structure learning algorithm *IBMAP-HC*, an efficient instantiation of the IBMAP approach of Eq. (4) that uses a closure based on Markov blankets, and a heuristic hill-climbing search to find the MAP structure $G^\star$ of Eq. (4).

3.1.1 Markov Blanket Closure

The *Markov blanket closure* is the closure set used by IBMAP-HC, consisting of a polynomial number of independence assertions (in the number of nodes of the graph). This closure is defined on the concept of *Markov blanket* $\mathbf{B}_X \subseteq \mathbf{V} \setminus \{X\}$ of a variable $X \in \mathbf{V}$ [16], defined as the set of all the nodes connected to X by an edge [26], i.e., its adjacency set.

Definition 2 (Markov blanket closure) The *Markov blanket closure* of a structure G is a set of independence assertions determined by the union of the Markov blanket closure $\mathcal{C}_X(G)$ of each variable X in the domain $\mathbf{V}$, i.e.,

$$\mathcal{C}(G) = \bigcup_{X \in V} \mathcal{C}_X(G), \tag{5}$$

where each $\mathcal{C}_X(G)$ is the union of two mutually exclusive sets of independence assertions:

$$\begin{aligned} \mathcal{C}_X(G) = &\Big\{ \langle X \not\perp\!\!\!\perp Y | \mathbf{B}_X \setminus \{Y\} \rangle \ : \ Y \in \mathbf{B}_X \Big\} \ \cup \\ &\Big\{ \langle X \perp\!\!\!\perp Y | \mathbf{B}_X \rangle \ : \ Y \notin \mathbf{B}_X \Big\}, \end{aligned} \tag{6}$$

that is, for each neighbor Y of X add a dependence assertion between both variables conditioned on the blanket of X, minus $\{Y\}$, and for each Y not a neighbor of X add an independence assertion between both variables conditioned on the blanket of X.

Appendix A presents a detailed proof that the Markov Blanket Closure is indeed a closure, that is, it completely determines the structure G used to construct it.

An interesting aspect of the Markov blanket closure proposed is that it results in a useful decomposition of the IB-score of Eq. (3), with important positive consequences in the efficiency of its computation:

$$\sigma(G) = \sum_{X \in \mathbf{V}} \sum_{i} \log \Pr(c_i \mid D). \tag{7}$$

By Eq. (6), the second summation in Eq. (7) has $(n-1)$ terms, one term per $Y \in \mathbf{V} \setminus \{X\}$. Each of these terms is denoted hereon by $\sigma_{X,Y}(G)$, and is computed as

$$\sigma_{X,Y}(G) = \begin{cases} \log \Pr(\langle X \not\perp\!\!\!\perp Y | \mathbf{B}_X \rangle \mid D) \text{ if } (X,Y) \text{ is an edge in } G \\ \log \Pr(\langle X \perp\!\!\!\perp Y | \mathbf{B}_X \rangle \mid D) \text{ otherwise} \end{cases}. \tag{8}$$

The decomposition of Eq. (7) can also be seen as a decomposition over variables, by grouping the second summation into the concept of *variable score* $\sigma_X(G)$. With this notation, Eq. (7) can be reformulated as

$$\sigma(G) = \sum_{X \in \mathbf{V}} \sigma_X(G) = \sum_{X \in \mathbf{V}} \sum_{Y \in \mathbf{V} \setminus \{X\}} \sigma_{X,Y}(G). \tag{9}$$

This decomposition allows an incremental computation of the score $\sigma(G')$ of some structure G' based on a previous computation of the score $\sigma(G)$ of a structure G, whenever G and G' differ by a constant number k of edges. For example, when G and G' differ by a single edge (X, Y), only the blankets of X and Y are affected, and therefore only independence assertions in $\mathcal{C}_X(G)$ and $\mathcal{C}_Y(G)$ involving these blankets are modified. Because of the decomposability of the score, only variable scores $\sigma_X(G)$ and $\sigma_Y(G)$ must be recomputed, with the possibility to reuse the $(n-2)$ variable scores $\sigma_W(G)$, for all $W \in \mathbf{V} \setminus \{X, Y\}$. This decomposition reduces the cost of computing the score for a structure G' from $O(n^2)$ to $O(n)$ tests, whenever the score of a neighbor structure G is already computed. For k edges differing between two structures at most $2k$ blankets are affected, with at most $2k$ variable scores requiring re-computation, again, a cost of $O(n)$ tests. Such incremental computation of the score has an important impact in local search optimization algorithms, such as the IBMAP-HC algorithm described in the next section, that proceeds by exploring successively structures that differs in one edge.

3.1.2 Our structure selection technique

To conclude the presentation of IBMAP-HC, we present a specific structure selection technique to perform the MAP search. The idea is to maximize the IB-score by a heuristic hill-climbing search in the space of structures, described in Algorithm 1.

Algorithm 1 IBMAP-HC (dataset D)

```
1: G ← empty structure with n nodes (columns in D)
2: current-score ← σ(G)
3: repeat
4:    G' ← select-next-structure(G, σ(G))
5:    neighbor-score ← σ(G')
6:    if neighbor-score < current-score then
7:       return G
8:    else
9:       G ← G'
10:      current-score ← neighbor-score

function select-next-structure(G, σ(G))
11: (X*, Y*) ← arg min_{(X,Y)∈G} σ_{X,Y}(G) + σ_{Y,X}(G)
12: return G with (X*, Y*) flipped
```

The algorithm has as input parameter the dataset D, used for computing the statistical independence tests. The search starts by computing the IB-score of a structure G with n nodes (number of columns in D) and no edges (lines 1 and 2), and then, in the main loop of line 3 iterates by selecting as the next structure G' in the search, the neighbor of G with maximum score (function *select-next-structure* called at line 4). In this algorithm we consider as neighbor structures all those structures differing by exactly one edge. The algorithm stops when the neighbor proposed does not improve the score, a condition checked at line 6. If the termination criteria is not reached, the variables G and *current-score* are assigned by the new structure and its score, which means that an ascent was made.

The optimal neighbor structure is obtained by the function *select-next-structure*. A naïve procedure would iterate over all $\binom{n}{2}$ neighbors, with a cost of $O(n^2)$ tests for computing the score for each, and a total cost of $O(n^4)$ tests. Instead, we propose here a heuristics that estimates the optimal neighbor without a single test computation, i.e., a cost of $O(1)$ test computations. Once the neighbor G' is selected, its score has to be computed (line 5) for comparing it against the score of G. This score can be computed incrementally, resulting in a total computational cost per iteration of $O(n)$ tests, a clear reduction from the $O(n^4)$ of the naïve procedure.

We proceed now to explain and justify the heuristics for selecting the optimal neighbor in the *select-next-structure* function. Being the neighbors of G all those structures $G_{X,Y}$ differing with G by an edge (X, Y), there is a one-to-one correspondence between each pair (X, Y) and each neighbor $G_{X,Y}$. The operation *flip* in line 12 consist in adding an edge to G if it does not exist, or remove it otherwise. Also, by Eq. (9) and the discussion of incremental computation of the score at the end of the previous section, each of these pairs contributes to the score of G independently of the others by $\sigma_X + \sigma_Y$, and this contribution is exactly the difference between the scores of G and $G_{X,Y}$. Therefore, the pair (X^*, Y^*) with the smallest contribution in G, results in a corresponding neighbor $G' = G_{X^*,Y^*}$ with the highest score among all neighbors. Performing the computation of $\sigma_X + \sigma_Y$ to get G' is equivalent to computing the score of G' incrementally with a cost of $O(n)$ tests, resulting in an overall cost of *select-next-structure* of $O(n^3)$ tests. The heuristics proposes an approximation that further reduces this computation, requiring not a single test computation.

The heuristics consists in approximating $\sigma_X(G)$ as $\sigma_{X,Y}(G)$, ignoring all other terms $\sigma_{X,W}(G)$, $W \in V \setminus \{X, Y\}$; and a similar approximation of $\sigma_Y(G)$ as $\sigma_{Y,X}(G)$. This heuristics is inspired by the fact that $|\sigma_{X,Y}(G) - \sigma_{X,Y}(G_{X,Y})| \gg |\sigma_{X,W}(G) - \sigma_{W,Y}(G_{X,Y})|$, for every W. Let us justify this. Since the edge (X, Y) is flipped between the two structures, it exists in one of them and does not exist in the other. Let us assume, without loss of generality, that it exists in G and does not exist in $G_{X,Y}$. Therefore, according to Eq. (8), the scores $\sigma_{X,Y}(G)$ and $\sigma_{X,Y}(G_{X,Y})$ of the two structures G and $G_{X,Y}$, respectively, evaluate to different cases: $\log \Pr(\langle X \not\perp\!\!\!\perp Y | \mathbf{B}_X \rangle \mid D)$ for the case of G, and $\log \Pr(\langle X \perp\!\!\!\perp Y | \mathbf{B}_X \rangle \mid D)$ for the case of $G_{X,Y}$. In contrast, for all other pair of variables (X, W), $W \in \mathbf{V} \setminus \{X\}$, either the edge (X, W) exists in both structures G and $G_{X,Y}$, or it does not exist in neither one. Therefore, their respective scores $\sigma_{X,Y}(G)$ and $\sigma_{X,Y}(G_{X,Y})$ both evaluates to the same case, either $\log \Pr(\langle X \not\perp\!\!\!\perp W | \mathbf{B}_X \rangle \mid D)$ if the edge exists, or $\log \Pr(\langle X \perp\!\!\!\perp Y | \mathbf{B}_X \rangle \mid D)$ if it does not. The only difference between these scores is in the blanket of X, that contains Y only in the case of G. This justifies that the difference between both scores is expected to be much larger for the case of $\{X, Y\}$ than for the case of $\{X, W\}$.

Since the exact score of the selected neighbor G' is computed in line 5, the only impact of the approximation in the hill-climbing search is the selection of a sub-optimal neighbor, which, although always climbs the search space, has the potential of producing a search termination before reaching a local maxima. Given the complexity of the problem, the impact of this approximation can only be assessed empirically. Later experiments show that despite this approximation, our approach still outperforms the state-of-the-art algorithms by reaching structures with higher quality. Moreover, Appendix B presents empirical measurements of the

complete landscape of the IB-score for several synthetic datasets, showing that in most cases, our structure selection strategy finds nearly optimal scores.

At this point the only aspect that remains to discuss is the resulting computational cost of the whole algorithm. The most expensive operation in the main loop becomes the computation of the (exact) IB-score of G' at line 5, with a cost of $O(n)$ when computed incrementally; resulting in an overall computational cost for the loop of $O(M \cdot n)$, where M denotes the number of iterations until termination. To this cost, it only remains to add the cost of computing the initial structure non-incrementally at line 1, with a cost of $O(n^2)$. Therefore, the overall cost of the algorithm is $O(n^2 + Mn)$. Since M can be obtained only empirically, the experimental section show measurements of M on different scenarios, proving empirically that M is not a source of an extra degree in the complexity, because it grows at most linearly with n, resulting in an overall computational complexity of $O(n^2)$.

4 Experimental results

This section describes several experiments on synthetic and real datasets for testing empirically the robustness of our approach IBMAP, and the efficiency of our algorithm IBMAP-HC. We report a detailed and systematic experimental comparison between IBMAP-HC and state-of-the art independence-based structure learning algorithms. For comparing all the algorithms on the same ground, we ran all of them using the Bayesian test as statistical independence test.

We compare the quality of structures learned by our solution, against the quality of structures learned by GSMN [9], a state-of-the-art independence-based algorithm in terms of quality. We introduce also a competitor called HHC-MN, as an adaptation for learning the structure of Markov networks of the HHC algorithm [5], a state-of-the-art independence-based algorithm for learning Bayesian networks.

The HHC algorithm learns the structure by learning the set of parents and children (PC) of each variable through the interleaved HITON-PC with symmetry correction algorithm [6, 4]. This is in fact possible for Bayesian networks, even though the Markov blanket of a variable is composed not only by the PC set, but also by the spouses of the variable (i.e., the other parents of its children). Interleaved HITON-PC executes at each iteration a step exponential in the size of the current estimate of the PC set. For the case of Markov networks, the equivalent of the PC of a variable is its neighbors, that is exactly its Markov Blanket. It is therefore expected that HITON-PC learns the Markov Blanket of a Markov network, and thus it can be used as part of HHC to learn the undirected structure. This fact is not proven analytically here, but confirmed empirically for all the cases considered in this section. To get a Markov network learning algorithm we then simply omit the final step of HHC that orients the edges, denoting the resulting algorithm by HHC-MN. As a final remark, we note that being the PC and Markov Blanket sets equivalent in Markov networks, the savings gained for Bayesian networks are non-existent and thus HHC-MN is expected to scale to fewer variables than its Bayesian networks counterpart.

The three following subsections describes our experiments with the competitor algorithms, over synthetic and real datasets.

4.1 Synthetic data experiments: random underlying structures

A first set of experiments were conducted on artificial datasets, generated using a Gibbs sampler on randomly generated Markov networks (structure plus parameters). This allows a systematic and controlled study, providing datasets with known underlying structures to allow the control of the complexity of the problem, and the ability to better asses the quality of structures obtained by each algorithm.

To measure structural errors in the structures learned for each algorithm, we report the *Hamming distance* between the learned structure and the underlying one, i.e., the sum of false positive and false negative edges of the learned structure.

Another quality measure that we use in this work for assessing the structures learned, is the well known F-measure, an harmonic mean of precision and recall quality measures, commonly used in the information retrieval community. Precision indicates how good was the algorithm in learning correct independences (that is, the relation between the true independences that were found, over all independences found by the algorithm). Instead, recall indicates how good was the algorithm in learning independences, but over all the correct independences present in the real structure (that is, the relation between the correct independences that were found, over the total of independences existent in the underlying structure). Then, the F-measure is computed as follows:

$$\text{F-measure} = \frac{2 \times precision \times recall}{precision + recall}. \tag{10}$$

The synthetic random Markov networks were generated for domains of $n \in \{75, 100, 200\}$ binary variables. For each domain size, 10 random networks were generated for increasing connectivities $\tau \in \{1, 2, 4, 8\}$, by considering as edges the first $n\tau/2$ variable pairs of a random permutation of the set of all variable pairs. It is worth mentioning that with increasing values of τ, it is increasingly difficult to learn the structure. Given these Markov networks, we report the quality of structures learned by GSMN, HHC-MN, and IBMAP-HC using portions of each dataset with increasing number of datapoints $D \in \{25, 50, 100, 200, 400, 800, 1600, 3200\}$, for each (n, τ) combination.

The independence structure determines the factorization of the distribution into potential functions over subset of variables, one per clique in the structure. To determine a complete model we must determine the numerical parameters that quantify these potential functions. For the datasets generated to correctly and strongly represent the direct dependencies encoded by the edges, we considered in these experiments pair-wise cliques for the factorization of the models, that is, two-variable factors $\phi(X, Y)$ for each edge in the random structure generated, and set the numerical parameters so that the correlation between them is strong. For that, we forced the parameters to result in a log-odds ratio of each pairwise factor $\varepsilon_{X,Y} = \log\left(\frac{\phi(X=0,Y=0)\phi(X=1,Y=1)}{\phi(X=0,Y=1)\phi(X=1,Y=0)}\right)$ to be equal to 1 for all edges (see [2]). This results in an equation over the values of the potential function with 4 unknowns. We therefore chose 3 parameters randomly in the range [0, 1], and solved for the remaining one. In our experiments we set $\varepsilon = 1.0$.

Figures 1 and 2 show the mean values and standard deviations over the ten repetitions, of the Hamming distances and F-measure of structures learned by the algorithms considered, respectively. The plots are ordered by columns for different n values, and by rows for different τ values. The figures show clearly that

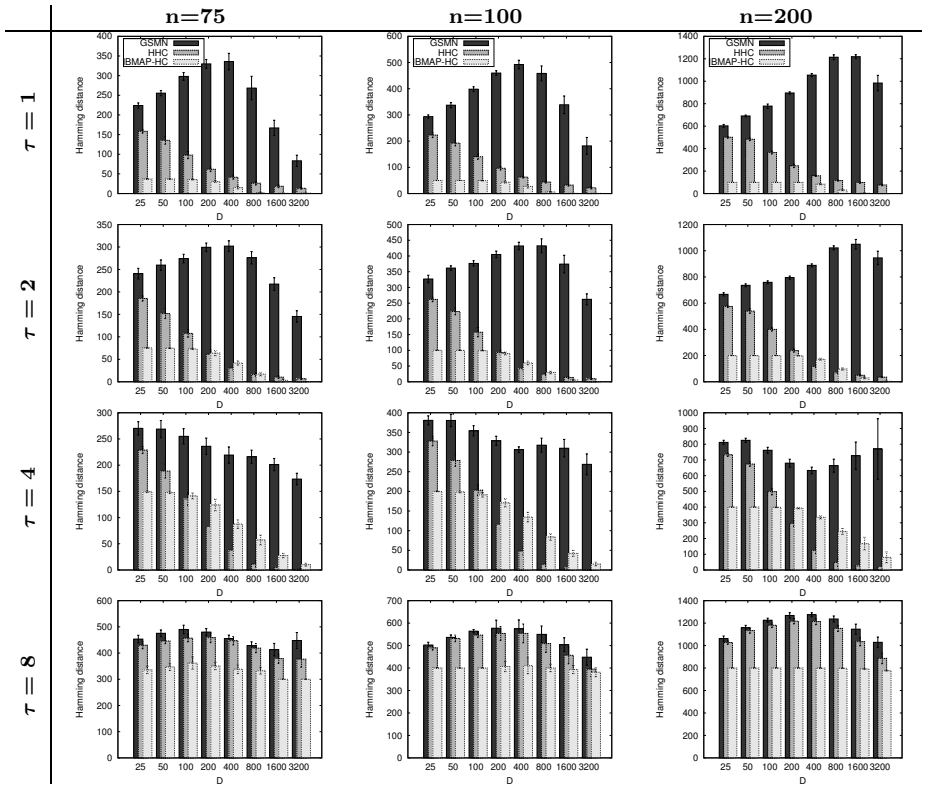


Fig. 1 Mean and standard deviation over 10 repetitions of the Hamming distance of the models learned by algorithms GSMN (black bars), HHC-MN (gray bars), and IBMAP-HC (light gray bars) for increasing sizes of random synthetic datasets, domain sizes $n = 75$ (left plot), $n = 100$ (center plot), and $n = 200$ (right plot), and $\tau \in \{1, 2, 4, 8\}$ in the rows.

both, IBMAP-HC and HHC-MN learn structures with qualities significantly better (lower Hamming distance, or higher F-measure) than that of GSMN in all the cases. With respect to HHC-MN, the quality of the structures learned by IBMAP-HC are better or equal (up to statistical significance) in all the cases tested, except in the following specific cases:

- $\tau = 2, n \geq 75, D = 400$,
- $\tau = 2, n = 200, D = 800$,
- $\tau = 4, n \geq 75, D \geq 200$.

The best improvements are obtained for $\tau = 1, D \in \{1600, 3200\}$ for all n's, where IBMAP-HC results in no errors, while GSMN and HHC-MN still present errors. Also it is worth noting that in all the cases considered, IBMAP-HC outperforms against competitors significantly in all cases (n,τ), when $D < 100$.

Figure 3 shows the corresponding running times of the same experiment. Such results show clearly that both, IBMAP-HC and HHC-MN runtimes are lower than that of GSMN in all cases, except some cases where HHC-MN presents expensive running times, due to its exponential cost for high connectivities (the cases with $\tau = 8$, in the last row).

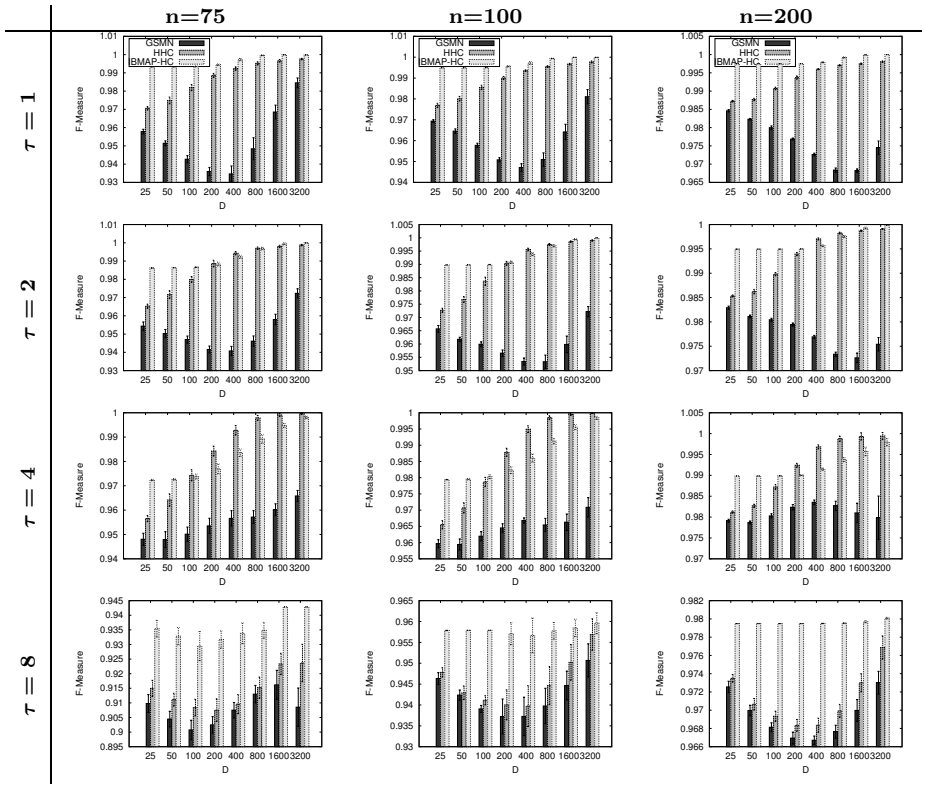


Fig. 2 Mean and standard deviation over 10 repetitions of the F-measure of the models learned by algorithms GSMN (black bars), HHC-MN (gray bars), and IBMAP-HC (light gray bars) for increasing dataset sizes of random synthetic datasets, domain sizes $n = 75$ (left plot), $n = 100$ (center plot), and $n = 200$ (right plot), and $\tau \in \{1, 2, 4, 8\}$ in the rows.

To conclude this section, we confirm empirically that IBMAP-HC achieves polynomial time complexities to the number of random variables in the domain. This is shown by Figure 4, that presents measurements of M (number of ascents in the hill-climbing search) for increasing problem sizes n. Such results were obtained for datasets generated in the same way than the previous experiments. The figure shows the values of M for problems with increasing values of $n \in \{4, 12, 16, 20, 24, 30, 50, 75\}$ in the X-axis, $D = 1000$, and a line for each $\tau \in \{1, 2, 4, 8\}$, indicating that M (Y-axis) grows linearly or slower. We omit results for different D values, because they are similar.

In summary, for synthetic datasets IBMAP-HC outperforms GSMN in quality in all cases, with equivalent runtimes, and outperforms HHC-MN in quality in most cases, with considerable improvements in runtime.

4.2 Synthetic data experiments: Ising models

A second set of experiments over synthetic datasets were conducted over a more interesting scenario, the Ising spin glasses models, that are mathematical models

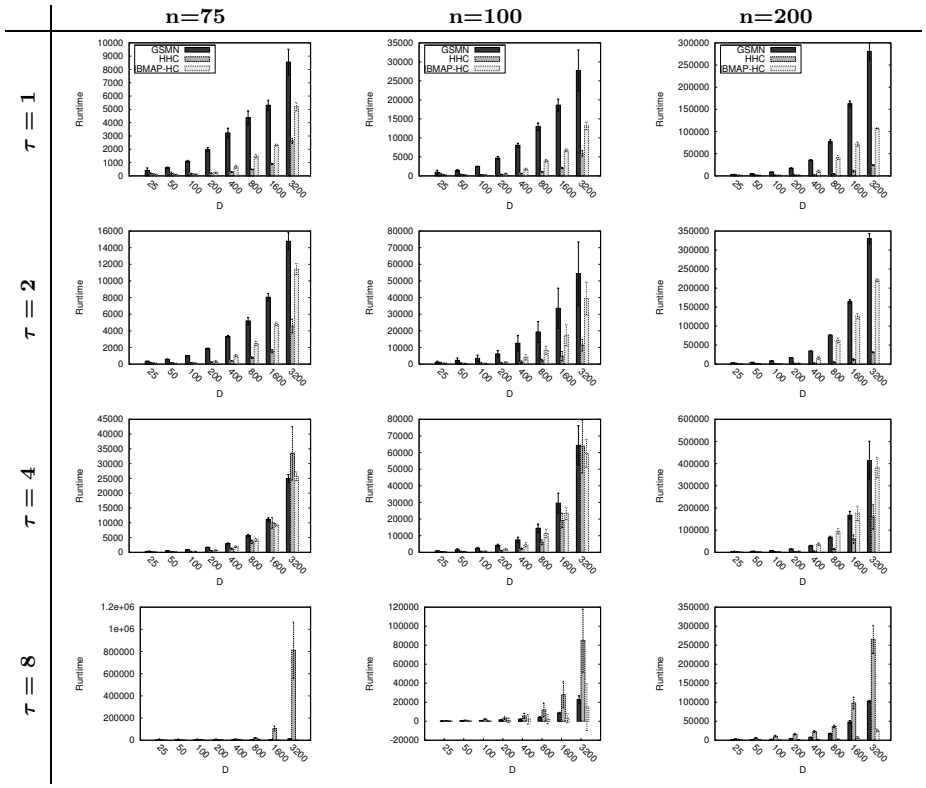


Fig. 3 Mean and standard deviation over 10 repetitions of the runtimes required by algorithms GSMN (black bars), HHC-MN (gray bars), and IBMAP-HC (light gray bars) for increasing dataset sizes of random synthetic datasets, domain sizes $n = 75$ (left plot), $n = 100$ (center plot), and $n = 200$ (right plot), and $\tau \in \{1, 2, 4, 8\}$ in the rows.

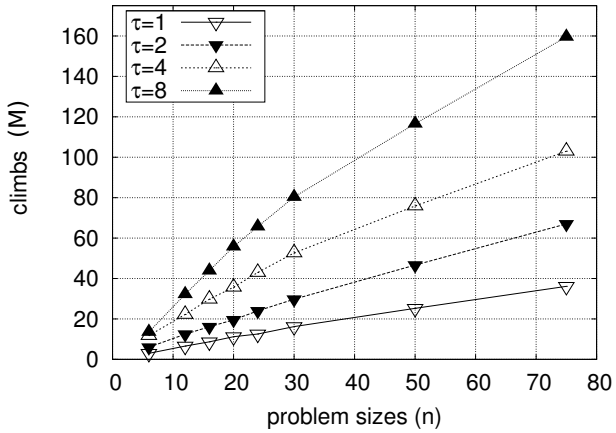


Fig. 4 Measurements in the number of ascents M (Y-axis) in the hill-climbing search of IBMAP-HC for increasing values of n (X-axis), and $\tau \in \{1, 2, 4, 8\}$, $D = 1000$.

of ferromagnetism in statistical mechanics. Using such models as underlying structure, ten datasets were generated for random Ising models with $n \in \{75, 100, 200\}$ binary variables. Figure 5 shows the results for ten different random repetitions. The graphs in such figure are ordered by columns for different n values, and showing in the first row the Hamming distance results, in the second row the F-measure results, and in the third row the corresponding runtimes. On the analysis of such results, we conclude they are similar to the case of random networks with $\tau = 2$, with IBMAP-HC outperforming GSMN and HHC-MN in all cases, in terms of Hamming distance, F-measure and runtimes.

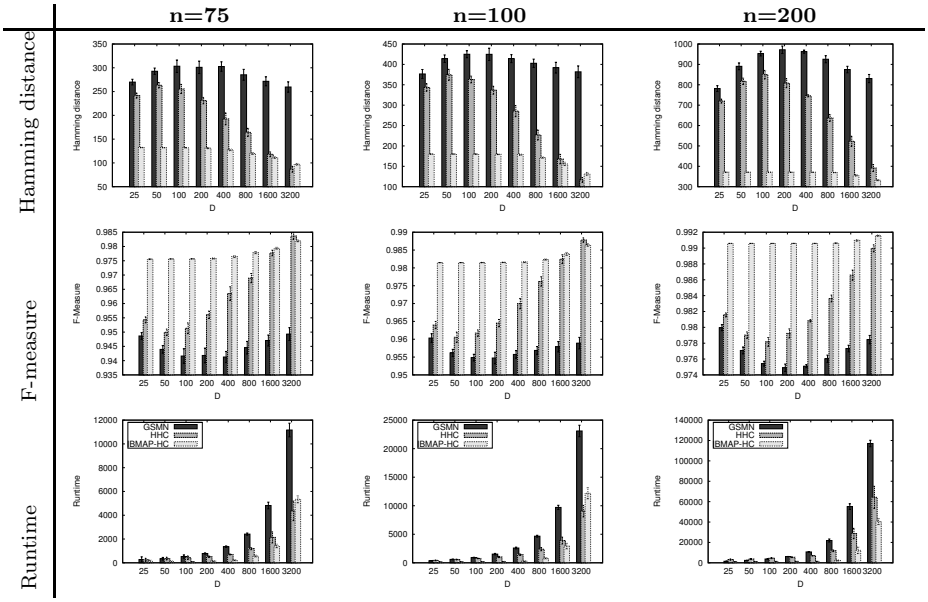


Fig. 5 Mean and standard deviation over 10 repetitions of the Hamming distance (first row), F-measure (second row) and runtime (third row) of algorithms GSMN (black bars), HHC-MN (gray bars), and IBMAP-HC (light gray bars) for increasing datasetsizes of Ising synthetic datasets, and domain sizes $n = 75$ (left plot), $n = 100$ (center plot), and $n = 200$ (right plot).

4.3 Benchmark datasets experiments

In this section we show our experiments on benchmark (real-world) datasets. We used the publicly available benchmark datasets obtained from the UCI Repositories of machine learning [1] and KDD datasets [15].

For benchmark datasets, since the underlying network is unknown, it is not possible to compute neither the Hamming distance nor the F-measure. Therefore, to measure the structure's quality we used a quantity called here *accuracy*, used for the same purpose in other related works [9, 21, 7]. The accuracy measure consists in a comparison of the outcome (true or false) of a number of tests performed on the structure learned by each algorithm using vertex separation, to the same tests

performed on the dataset. We define the accuracy as a normalized measure for counting the number of matches in a comparison of the independence assertions that holds in data and the structure learned. That is, if $\mathcal{T}$ denotes the set of all possible triplets over $\mathbf{V}$, it is checked for how many triplets $t \in \mathcal{T}$, t is independent (or dependent) in both the data, and the learned structure, and then normalized by $|\mathcal{T}|$. Unfortunately, the size of $\mathcal{T}$ is exponential, so we compute the approximate accuracy over a randomly sampled subset $\widehat{\mathcal{T}}$, uniformly distributed for each conditioning set cardinality. In our experiments we used $|\widehat{\mathcal{T}}| = 100 \times \binom{n}{2}$, i.e., a hundred triplets per conditioning set size.

We conducted our experiment using 19 real-world domains, listed in Table 1, column one. For each dataset D, we shuffled the data and then divided it into a training set for learning the structure (%75), and a test set for computing the accuracy (%25). The table also shows information about the number of attributes (second column), and the number of datapoints available in the train and test sets (third and fourth column). For each dataset we used the train set as input to the GSMN, HHC-MN, and IBMAP-HC algorithms, and the accuracy obtained for the structure learned for each algorithm is shown in the fifth, sixth and seventh columns, respectively. For each evaluation measure, the best performance is indicated in bold. Such results show that in 8 out of 17 datasets IBMAP-HC resulted in better accuracy, 6 cases resulted in ties (2 with GSMN, 1 with HHC-MN, and 3 with both), and for the remaining cases, the best results are obtained by HHC-MN(2 cases) and GSMN (1 case).

		Train	Test	**Accuracy**		
Dataset	n	D	D	**GSMN**	**HHC-MN**	**IBMAP-HC**
machine	10	156	52	0.590	0.567	**0.679**
lenses	5	17	6	**0.881**	0.875	**0.881**
hepatitis	20	59	20	0.496	0.633	**0.796**
hayes-roth	6	98	33	**0.516**	**0.516**	**0.516**
crx	16	489	163	0.578	0.593	**0.609**
cmc	10	1104	368	**0.759**	0.711	0.726
car	7	1295	432	0.629	0.641	**0.703**
bands	38	207	69	0.399	0.408	**0.546**
baloons	5	14	5	**0.950**	0.897	**0.950**
balance-scale	5	468	156	**0.516**	**0.516**	**0.516**
nursery	9	9719	3240	0.392	0.415	**0.649**
ecoli	9	251	84	0.523	0.591	**0.694**
echocardiogram	13	45	15	0.696	**0.745**	**0.745**
flag	29	145	48	0.446	0.451	**0.803**
iris	5	112	37	0.695	**0.742**	0.736
tic-tac-toe	10	718	239	0.671	**0.684**	0.498
monks-1	7	416	139	**0.905**	**0.905**	**0.905**

Table 1 Accuracy for several real-world data sets. The structure is learned using a subsample called train set, and the accuracy is computed using the test set. For each evaluation measure, the best performance is indicated in bold.

5 IBMAP-HC for Estimation of Distribution Algorithms

In contrast to benchmark datasets that comes from arbitrary applications, we present now results of evaluating IBMAP-HC in a real world application of knowledge-discovery: the *Estimation of Distribution algorithms* (EDAs) [25,17]. These are variations of the well-known evolutionary algorithms, that perform the same *selection* and *variation* stages, but replace the *crossover* and *mutation* stages with the *estimation* and *sampling* in the task of generating a new population. The former stage *estimate* a probability distribution from the current population, generating the next population by *sampling* from it (thus their name). In the *estimation* stage, EDAs estimate the probability distribution from the dataset corresponding to the current population. This is because they associate each gene to a random variable, each individual to a joint assignment of these variables, and the selected population to a sample of the distribution. The rationale for replacing crossover methods with estimation is that by estimating the distribution from the selected individuals, that is, those best fitted, the sampling stage would produce novel, yet well-fitted individuals.

Several Markov networks based EDAs has been proposed recently that uses Markov networks for modeling the distribution [28,3,30,31]. As a test-bed we considered the *Markovianity Optimization Algorithm* (MOA) [31]. This is a state-of-the-art MN-based EDA that learns the Markov network structure from the population using an efficient structure learning algorithm based on mutual information (MI), a simple independence-based structure learning algorithm, described in detail in the same work, and designed specifically for MOA. The sampling in MOA is conducted through a variation of a Gibbs sampler that requires only the structure of the model, avoiding the need to learn the model parameters. The implementation of MI in MOA takes advantage of experts information indicating the maximum number of neighbor variables that a variable can have, denoted here k. We tested MI for different values of k (results not shown here), observing great sensitivity of MI to its value. Our algorithm IBMAP-HC does not use such parameter. In the experiments below we set the value of k for MI to be the closest to the true value, resulting in the best possible performance of MI, i.e., the strongest competitor for IBMAP-HC.

	MOA		**MOA'**	
n	D^*	f^*	D^*	f^*
15	50	267.50 (35.45)	50	202.50 (14.19)
30	200	1170.00 (94.87)	100	475.00 (42.49)
60	800	5200.00 (98.46)	200	1050.00 (52.70)
90	800	5560.00 (126.49)	400	2220.00 (63.25)
120	1600	11200.00 (871.53)	800	4400.00 (312.33)

Table 2 Results of MOA and MOA' (that uses IBMAP-HC) for the OneMax problem, for increasing problem sizes (rows) in terms of critical population size D^*, and mean and standard deviation over 10 repetitions of the number of fitness evaluations f^* required to obtain the global optimum.

We conducted experiments to compare IBMAP-HC as an alternative structure learning within MOA, denoted MOA$'$, and denoting by MOA the original version

that uses MI. The thesis is that a better structure learning algorithm improves the convergence of MOA, that is, the optimum is reached computing fewer evaluations of the fitness of individuals. Both versions were tested on two benchmark functions widely used in the EDA's literature: *Royal Road* and *OneMax*, both bit-string optimization tasks, detailed in [24]. Each bit-string is modeled in the context of evolutionary algorithms as a chromosome and each bit as a gene. In the Royal Road problem, the variables are arranged in groups of size γ. Its goal is to maximize the number of 1s in the string, but adding γ to the fitness count only when a group has all 1s, otherwise adding 0. For example, in the case of $\gamma = 4$, an individual 111110011111 is separated in the groups [1111] [1001] [1111], and only the first and third groups contributes 4 to the fitness count, which in the example equals 8. The underlying independence structure that should be learned therefore contains cliques of size γ, one per group. In our experiments we used $\gamma = 1$ and $\gamma = 4$. The former is known in the literature as *OneMax*. In the example, the fitness is 10 for OneMax. Clearly, the optimal individual for both problems is 111111111111.

	MOA		MOA'	
n	D^*	f^*	D^*	f^*
16	100	545.00 (59.86)	50	337.50 (176.09)
32	400	3800.00 (210.82)	400	2140.00 (134.99)
64	800	9120.00 (252.98)	800	4440.00 (126.49)
92	1600	18400.00 (533.33)	800	5080.00 (500.67)
120	1600	31120.00 (822.31)	1600	9840.00 (386.44)

Table 3 Results of MOA and MOA' (that uses IBMAP-HC) for the Royal Road problem, for increasing problem sizes (rows) in terms of critical population size D^*, and mean and standard deviation over 10 repetitions of the number of fitness evaluations f^* required to obtain the global optimum.

In the experiments, MOA is iterated for 1000 generations or until the optima is reached, whatever happened first. For several runs differing in the initial (random) population, we measured the *success rate* as the fraction of times the optima is found. A commonly used measure of performance in EDAs is the *critical population size* D^*; the minimum population size for which the success rate is 100%. Smaller D^* values have a double benefit over runtime: (i) fewer fitness evaluations for reaching the optima, and (ii) faster distribution estimation. We report D^* and the number of fitness evaluations required for that population size, denoted f^*. More robust algorithms are expected to require smaller D^* and f^* values. To measure D^* in Royal Road and OneMax, each version of MOA was run 10 times for each of the population sizes $D = \{50, 100, 200, 400, 800, 1600, 3200\}$. Then, for that D^*, we report the average and standard deviation of f^* on each of those runs. In all the experiments, the population is truncated with a selection size of 50% and an elitism of 50%; used for preventing diversity loss. In MOA, the parameter k was set to 3 and 1 in Royal Road and OneMax, respectively.

Results are presented in Table 2 for the OneMax problem, and Table 3 for the Royal Road problem. For both algorithms MOA and MOA′, each table report values of D^*, and the average and standard deviation (in parenthesis) of f^*, for increasing problem sizes $n = \{15, 30, 60, 90, 120\}$ for the OneMax problem, and $n = \{16, 32, 64, 90, 120\}$ for the Royal Road problem (domains multiple of $\gamma = 4$

are required). In both tables, the results show that for f^*, MOA$'$ always outperforms MOA; while for D^*, it is always equal or lower. For Royal Road, the larger improvement is for $n = 92$ where MOA$'$ requires 75% fewer fitness evaluations f^* and D^* is halved. For OneMax, the larger improvement is for $n = 60$ where MOA$'$ requires 80% fewer fitness evaluations f^* and D^* is reduced to a quarter.

An interpretation of these results is that IBMAP-HC estimates better the distribution. To confirm this hypothesis we compared the structures learned by the two algorithms over the same synthetic datasets considered in the previous section. For $n = 75$, $D = 100$, $\tau = 2$, the Hamming distances of MI and IBMAP-HC were 132, and 75, respectively. For $\tau = 4$ they were 233 and 143, respectively; and for $\tau = 8$, 395 and 388, respectively. These results show clearly that the quality of IBMAP-HC indeed outperforms that of MI. Finally, we highlight that the efficiency of IBMAP-HC allowed it to be run in large problems up to 120 genes in size, estimating the structure over many generations.

6 Conclusions and future work

This paper proposes a novel independence-based, maximum-a-posteriori approach for learning the structure of Markov networks; and IBMAP-HC, an efficient instantiation of IBMAP. Our method follows an independence-based strategy for getting the MAP independences structure from data proposing an independence-based score. Experiments comparing IBMAP-HC against state-of-the-art independence-based algorithms indicate that our method improves in most cases over the independence-based competitors with equivalent computational complexities. IBMAP-HC was also tested in a practical, challenging setting: Estimation of Distribution algorithms, resulting in faster convergence to the optimum than a state-of-the-art Markov network EDA algorithm, for the selected benchmark functions. According with our experimental results, and the conclusions of Appendix B, the effectiveness of our structure selection strategy is confirmed, and therefore we believe that it is worth guiding our future work in improving the IB-score as a measure of $\Pr(G \mid D)$, i.e., relaxing the independence assumption made in Equation (3), as well as exploring alternative closure sets. Also, it is clearly worthwhile considering testing our approach in more practical real world testbeds, potentially comparing its performance against state-of-the-art score-based algorithms, such as [14, 27, 12, 33].

7 Acknowledgements

This work was funded by the grant PICT-241 of the National Agency of Scientific and Technological Promotion, FONCyT, Argentina; the grant PID-1205 of the National Technological University, Argentina; and the scholarship program for teachers of the National Technological University and the Ministry of Science, Technology and Productive Innovation; Argentina. Special thanks to Roberto Santana and Siddartha Shakya for their help and support while implementing our experiments on EDAs.

A Completeness of Markov blanket closure

This appendix presents Theorem 1, a formal proof that the Markov blanket closure described in Definition 2 of Section 3.1.1 is in fact a closure, i.e., its independence assertions completely determine the structure used to generate it.

Let us start by reproducing some necessary theoretical results extracted from [16, 18, 26]: the *pairwise Markov property*, the *Intersection property* of conditional independence, and the *Strong Union property* of conditional independence, all satisfied by any Markov network G of a positive graph-isomorph distribution P:

Definition 3 (Pairwise Markov property) Let G be a Markov network of some graph-isomorph distribution P, then

$$(X, Y) \notin E(G) \Leftrightarrow \langle X \perp\!\!\!\perp Y | V \setminus \{X, Y\} \rangle \text{ in } P. \tag{11}$$

Definition 4 (Intersection) The conditional independences among random variables of a positive distribution P satisfy the *Intersection* property (expressed in counter-positive form):

$$\langle X \not\perp\!\!\!\perp Y | \mathbf{Z} \rangle \wedge \langle X \perp\!\!\!\perp W | \mathbf{Z}, Y \rangle \Rightarrow \langle X \not\perp\!\!\!\perp Y | \mathbf{Z}, W \rangle \tag{12}$$

for all $(X \neq Y \neq W) \notin \mathbf{Z}$.

Definition 5 (Strong Union) The conditional independences among random variables of a graph-isomorph distribution P satisfy the following *Strong Union* property of conditional independence:

$$\langle X \perp\!\!\!\perp Y | \mathbf{Z} \rangle \Rightarrow \langle X \perp\!\!\!\perp Y | \mathbf{Z}, W \rangle \tag{13}$$

for all $(X \neq Y) \notin \mathbf{Z}$.

We present now two auxiliary lemmas that relate independences with edges in the graph:

Lemma 1

$$\langle X \perp\!\!\!\perp Y | \mathbf{B}_X \setminus \{Y\} \rangle \Rightarrow \ (X, Y) \notin E(G). \tag{14}$$

Proof. The proof proceeds by first applying the Strong union property to the l.h.s. to obtain $\langle X \perp\!\!\!\perp Y | \mathbf{V} \setminus \{X, Y\} \rangle$, and then applying the pairwise property to conclude the r.h.s. $(X, Y) \notin E(G)$. □

For the remaining of the proof we need to argue that something similar to the counter-positive of Lemma 1 holds:

Lemma 2

$$\langle X \not\perp\!\!\!\perp Y | \mathbf{B}_X \setminus \{Y\} \rangle \ \wedge \ \forall W \notin \mathbf{B}_X \langle X \perp\!\!\!\perp W | \mathbf{Z}, Y \rangle \Rightarrow \ (X, Y) \in E(G). \tag{15}$$

Proof. The proof proceeds by extending the conditioning set $\mathbf{B}_X \setminus \{Y\}$ of the l.h.s. to the whole domain $V \setminus \{X, Y\}$, to then apply the counter-positive of Eq. (11) and reach the r.h.s. $(X, Y) \in E(G)$. For that, we apply the intersection property of Eq. (12) iteratively, by taking at each iteration the pair containing one of the independences in the l.h.s., and, in the first iteration the dependence in the l.h.s., and the following iterations the dependence resulting from applying intersection. In all cases, we take $\mathbf{Z} = \mathbf{B}_X \setminus \{Y\}$. Let see this process in detail. In the first iteration we take from the l.h.s. the dependence and the independence for the first W, obtaining, by intersection, the dependence $\langle X \not\perp\!\!\!\perp Y | \mathbf{Z}, W \rangle$. We can now take the resulting dependence, with the independence for the following W, denoted for convenience W'. It seems that intersection can no longer be applied because the respective conditioning sets $\mathbf{Z} \cup \{W\}$ and $\mathbf{Z} \cup \{Y\}$ does not match. However, by graph-isomorphism of P, we have that the *Strong Union* property of conditional independence is satisfied in P, and therefore any independence given some conditioning set follows from the same independence given a subset of this conditioning set, in particular then, we have that $\langle X \perp\!\!\!\perp W' | \mathbf{Z}, W, Y \rangle$, and intersection can therefore be applied, resulting in $\langle X \not\perp\!\!\!\perp Y | \mathbf{Z}, W, W' \rangle$. Following this iteratively, we reach $\langle X \not\perp\!\!\!\perp Y | V \setminus \{X, Y\} \rangle$, where the conditioning set is the result of $\mathbf{Z} = \mathbf{B}_X \setminus \{Y\} \cup \mathbf{B}_X$, recalling $X \notin \mathbf{B}_X$.

□

We can now prove our main theorem:

Theorem 1 *The* Markov blanket closure *of a structure G, as stated in Definition 2, is a set of conditional independence assertions that are sufficient for completely determining the structure G of a positive graph-isomorph distribution.*

Proof. We prove the above theorem by proving that all the edges and no edges in G are determined by the assertions contained in $\mathcal{C}(G)$. We do it separately for absence and existence of edge between any two variables X and Y:

i) **For edge absence:** Let $(X, Y) \notin E(G)$. Then, by definition, the closure contains the two independence assertions: $\langle X \perp\!\!\!\perp Y | \mathbf{B}_X \setminus \{Y\} \rangle$ and $\langle Y \perp\!\!\!\perp X | \mathbf{B}_Y \setminus \{X\} \rangle$, which, by Eq. (14) of Lemma 1 both imply $(X, Y) \notin E(G)$.

ii) **For edge existence:**
Similarly, let $(X, Y) \in E(G)$. Then, by definition, the closure contains the dependence assertion: $\langle X \not\perp\!\!\!\perp Y | \mathbf{B}_X \setminus \{Y\} \rangle$. Also, for all W s.t. $(X, W) \notin E(G)$ (i.e., $W \notin \mathbf{B}_X$), the closure contains $\langle X \perp\!\!\!\perp W | \mathbf{B}_X \rangle$. Then, by Eq. (15) of Lemma 2 we have that $(X, Y) \in E(G)$. □

B IBMAP landscape analysis

In this appendix we report the results of an experiment that analyzes empirically the landscape of the IB-score function on synthetic datasets. The experiment consists in an analysis of the surface of the IB-score over the complete search space of possible structures. The aim is to assess how good is the hill-climbing search for maximizing the IB-score. Due to the exponential number of possible networks for each domain, in a first instance we explore how the complete landscape of IB-score looks like for datasets with a small domain size $n = 6$. For this experiment, we used synthetic datasets similar to those used in Section 4.1.

The plots in Figure 6 show in the Y-axes the values of the IB-score for all the possible structures, and sort the structures in the X-axes, by its Hamming distance to the true underlying structure in the dataset (this is, from zero, to $\binom{n}{2}$). Note that the scores of the structures appear in log probabilities, because they was computed as shown in Equation (3). With this layout, the structures in the left (near to zero) are those with less structural errors, and are also those expected to have a higher value of the IB-score. Therefore, the structures in the right are expected to have lower values of the IB-score. Also, indicated with a diamond, the structures found by the algorithm IBMAP-HC are shown for each case.

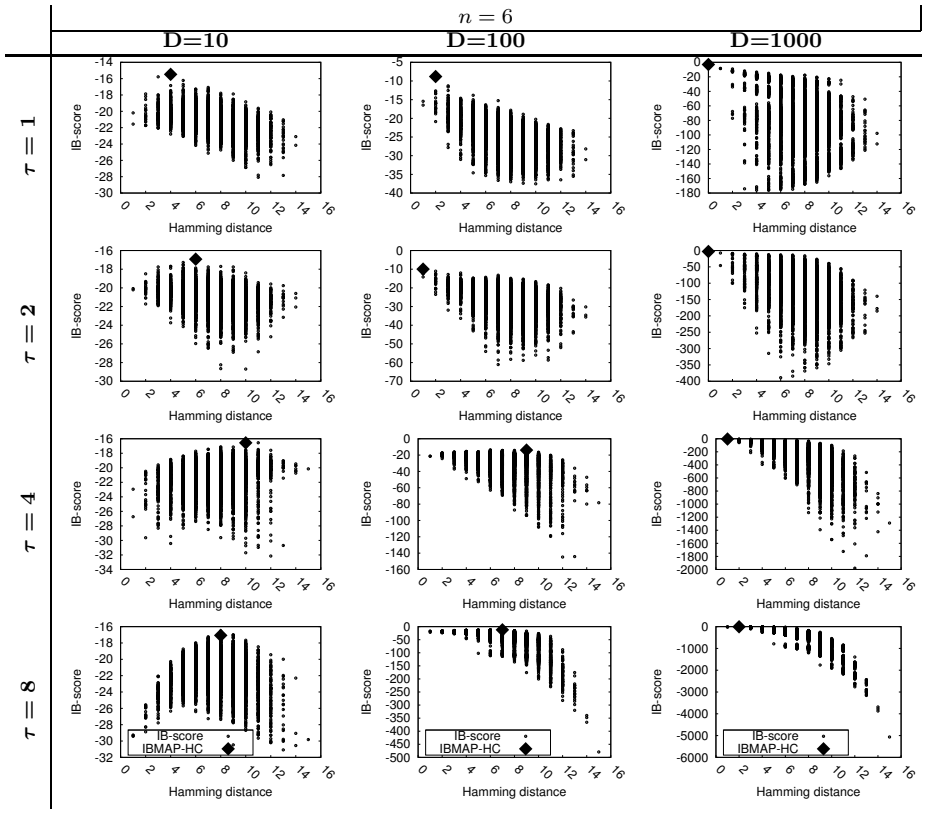


Fig. 6 Complete landscape of the IB-score for synthetic datasets with $n = 6$, for increasing dataset sizes $D = 10$ (left column), $D = 100$ (center column), and $n = 1000$ (right column), and $\tau \in \{1, 2, 4, 8\}$ in the rows. The X-axis sort the structures in the Hamming distance with the correct structure. The Y-axis shows the IB-score for all the structures in the landscape. The structure found by IBMAP-HC is indicated by a diamond.

The plots are ordered in the columns for increasing values of the dataset $D \in \{10, 100, 1000\}$, and in the rows, the different values of $\tau \in \{1, 2, 4, 8\}$, increasing the complexity of the problem. From the analysis of such plots, it is observed how the landscape shapes to a decreasing curve as increasing the value D (see the tendency from left to right columns). This is achieved because the precision of the statistical tests improves with increasing D. In second place, the diamond that indicates the position in the landscape of the structure learned by the IBMAP-HC algorithm, achieves always the structure with highest score value. It can be also observed how the error of the structure learned by IBMAP-HC is closer to zero while increasing D.

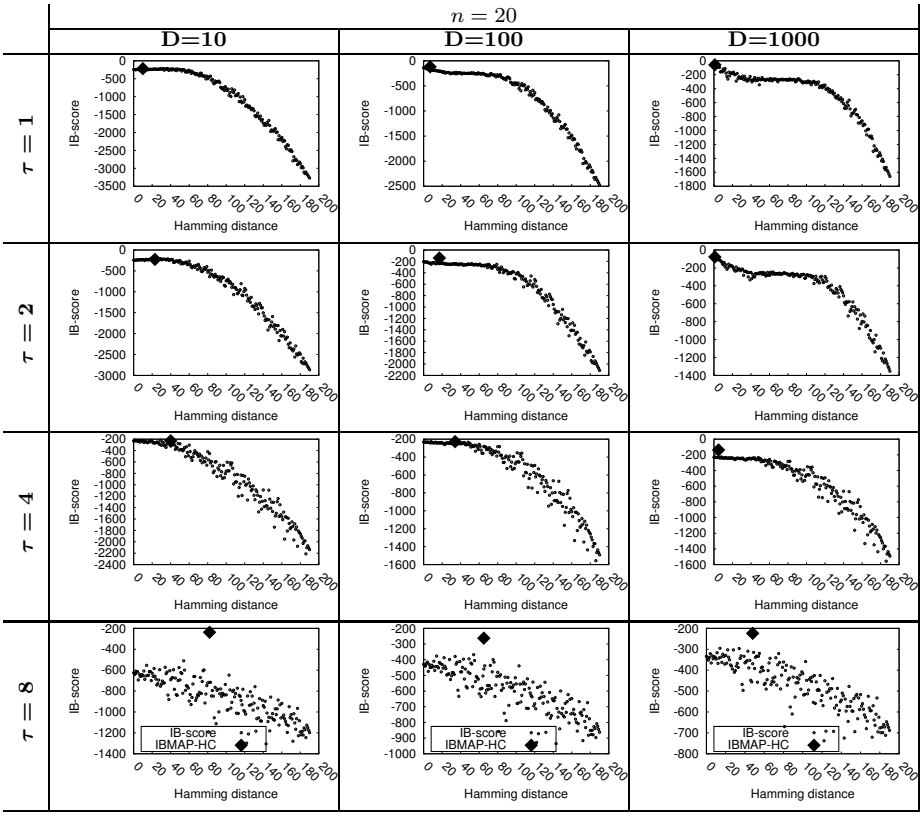


Fig. 7 A fraction of the landscape of the IB-score for synthetic datasets with $n = 20$, for increasing dataset sizes $D = 10$ (left column), $D = 100$ (center column), and $n = 1000$ (right column), and $\tau \in \{1, 2, 4, 8\}$ in the rows. The X-axis sort the structures in the Hamming distance with the correct structure. The Y-axis shows the IB-score for all the structures in the landscape. The structure found by IBMAP-HC is indicated by a diamond.

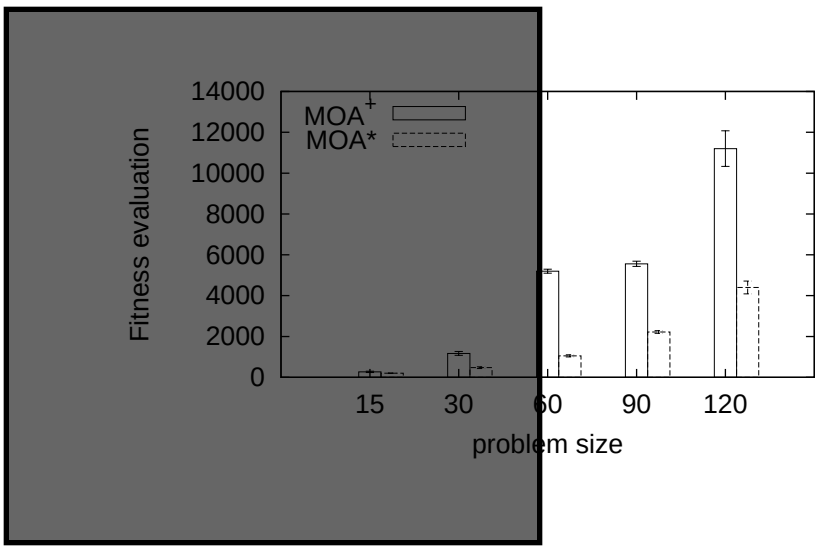
A second instance of this experiment was made for a domain size $n = 20$. In this instance, the landscape contains a total size of $2^{\binom{20}{2}}$. As it is impossible to show the IB-score for the complete landscape, we show only a subset obtained by generating randomly $k = 5$ structures deferring in m edges with the true structure, with m from 0 to $\binom{20}{2}$ in the X-axis. Such results are shown in Figure 7. From the analysis of such plots, the same conclusions are observed.

To conclude this appendix, it is worth noting that our results confirm the effectiveness of our structure selection strategy in maximizing the IB-score over the complete landscape. For that reason, we conclude that it is worth guiding our future work only in the improvement of the IB-score as a measure of $\Pr(G \mid D)$.

References

1. A. Asuncion, D.N.: UCI machine learning repository (2007)
2. Agresti, A.: Categorical Data Analysis, 2nd edn. Wiley (2002)
3. Alden, M.: MARLEDA: Effective Distribution Estimation Through Markov Random Fields. Ph.D. thesis, Dept of CS, University of Texas Austin (2007)
4. Aliferis, C., Statnikov, A., Tsamardinos, I., Mani, S., Koutsoukos, X.: Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part I: Algorithms and Empirical Evaluation. JMLR **11**, 171–234 (2010)
5. Aliferis, C., Statnikov, A., Tsamardinos, I., Mani, S., Koutsoukos, X.: Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part II: Analysis and Extensions. JMLR **11**, 235–284 (2010)
6. Aliferis, C., Tsamardinos, I., Statnikov, A.: HITON, a novel Markov blanket algorithm for optimal variable selection. AMIA Fall (2003)
7. Bromberg, F., Margaritis, D.: Improving the Reliability of Causal Discovery from Small Data Sets using Argumentation. JMLR **10**, 301–340 (2009)
8. Bromberg, F., Margaritis, D., Honavar, V.: Efficient markov network structure discovery using independence tests. In: In Proc SIAM Data Mining, p. 06 (2006)
9. Bromberg, F., Margaritis, D., V., H.: Efficient Markov Network Structure Discovery Using Independence Tests. JAIR **35**, 449–485 (2009)
10. Chickering, D.M.: Learning Bayesian networks is NP-Complete. In: D. Fisher, H. Lenz (eds.) Learning from Data: Artificial Intelligence and Statistics V, pp. 121–130. Springer-Verlag (1996)
11. Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley-Interscience, New York, NY, USA (1991)
12. Davis, J., Domingos, P.: Bottom-Up Learning of Markov Network Structure. In: ICML, pp. 271–278 (2010)
13. Della Pietra, S., Della Pietra, V.J., Lafferty, J.D.: Inducing Features of Random Fields. IEEE Trans. PAMI. **19**(4), 380–393 (1997)
14. Ganapathi, V., Vickrey, D., Duchi, J., Koller, D.: Constrained Approximate Maximum Entropy Learning of Markov Random Fields. In: Uncertainty in Artificial Intelligence, pp. 196–203 (2008)
15. Hettich, S., Bay, S.D.: The UCI KDD archive (1999)
16. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
17. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. Kluwer Pubs (2002)
18. Lauritzen, S.L.: Graphical Models. Oxford University Press (1996)
19. Lee, S.I., Ganapathi, V., Koller, D.: Efficient structure learning of Markov networks using L1-regularization. In: NIPS (2006)
20. Margaritis, D.: Distribution-Free Learning of Bayesian Network Structure in Continuous Domains. In: Proceedings of AAAI (2005)
21. Margaritis, D., Bromberg, F.: Efficient Markov Network Discovery Using Particle Filter. Comp. Intel. **25**(4), 367–394 (2009)
22. McCallum, A.: Efficiently inducing features of conditional random fields. In: Proceedings of Uncertainty in Artificial Intelligence (UAI) (2003)
23. Minka, T.: Divergence measures and message passing. Tech. rep., Microsoft Research (2005)
24. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA, USA (1998)
25. Mühlenbein, H., Paaß, G.: From recombination of genes to the estimation of distributions I. binary parameters. In: H.M. Voigt, W. Ebeling, I. Rechenberg, H.P. Schwefel (eds.) Parallel Problem Solving from Nature PPSN IV, *Lecture Notes in Computer Science*, vol. 1141, pp. 178–187. Springer Berlin / Heidelberg (1996). 10.1007/3-540-61723-X_982
26. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc. (1988)
27. Ravikumar, P., Wainwright, M.J., Lafferty, J.D.: High-dimensional Ising model selection using L1-regularized logistic regression. Annals of Statistics **38**, 1287–1319 (2010). DOI 10.1214/09-AOS691
28. Santana, R.: Estimation of distribution algorithms with kikuchi approximations. Evol. Comput. **13**(1), 67–97 (2005). DOI 10.1162/1063656053583496. URL `http://dx.doi.org/10.1162/1063656053583496`

29. Schlüter, F.: A survey on independence-based markov networks learning. Artificial Intelligence Review pp. 1–25 (2012). URL http://dx.doi.org/10.1007/s10462-012-9346-y. 10.1007/s10462-012-9346-y
30. Shakya, S., McCall, J.: Optimization by estimation of distribution with deum framework based on markov random fields. International Journal of Automation and Computing **4**(3), 262–272 (2007). URL http://www.springerlink.com/index/10.1007/s11633-007-0262-6
31. Shakya, S., Santana, R., Lozano, J.A.: A markovianity based optimisation algorithm. Genetic Programming and Evolvable Machines **13**(2), 159–195 (2012)
32. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. Adaptive Computation and Machine Learning Series. MIT Press (2000)
33. Van Haaren, J., Davis, J.: Markov network structure learning: A randomized feature generation approach. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (2012). URL https://lirias.kuleuven.be/handle/123456789/345604
34. Welsh, D.J.A.: Complexity: knots, colourings and counting. Cambridge University Press, New York, NY, USA (1993)


14000
12000
10000
8000
6000
4000
2000
0
Fitness evaluation
MOA+
MOA*
15
30
60
90
120
problem size